John von Neumann Institute for Computing

PSO@Autodock : A Novel Bio-Algorithm-Based Fast Flexible Docking Tool for Virtual Screening

V. Namasivayam, R. Günther

published in

From Computational Biophysics to Systems Biology (CBSB07),
Proceedings of the NIC Workshop 2007,
Ulrich H. E. Hansmann, Jan Meinke, Sandipan Mohanty,
Olav Zimmermann (Editors),
John von Neumann Institute for Computing, Jülich,
NIC Series, Vol. **36**, ISBN 978-3-9810843-2-0, pp. 239-241, 2007.

http://www.fz-juelich.de/nic-series/volume36

PSO@Autodock : A Novel Bio-Algorithm-Based Fast Flexible Docking Tool for Virtual Screening

Vigneshwaran Namasivayam and Robert Günther

Institute of Biochemistry, University of Leipzig,
Brüderstrasse 34, 04103 Leipzig, Germany
E-mail: {vignesh, robguent}@uni-leipzig.de

Virtual Screening (VS) has become an essential element in the drug discovery. Based on the 3D structure of the receptor, molecular docking studies can be applied to screen large libraries of compounds for candidate molecules specific for the particular target. Molecular docking simulation methods describe protein-ligand interactions at atomic level within a force field-based approach. However, due to their high computational costs, molecular docking simulation methods have rarely been applied to VS. We present PSO@Autodock, a novel molecular docking molecular tool for virtual screening. It is based on Particle Swarm Optimization (PSO) and has been implemented in AutoDock3 (AD3), a widely used docking program. In comparison to standard AD3, PSOAutodock requires less than 20% of computing time for predicting the correct protein-ligand complex. Thus, simulating the docking of 2000 compounds with full flexible treatment of the ligands employing PSO@Autodock can now be finished over night on a standard personal computer.

1 Introduction

Virtual Screening (VS) techniques applying computational molecular docking methods have proven to be a viable alternative to experimental High Throughput Screening (HTS). Given the 3-D structure of the protein target, molecular docking methods allow screening large libraries of compounds and provide detailed information on the protein-ligand interactions.

Currently, most VS strategies employ geometric matching methods that dock an ensemble of ligands and rank them based on empirical scoring functions. On contrary, simulation methods apply force fields to describe the molecular docking process in more detail. However, this is computational demanding. Simulation methods rely on the assumption that the protein-inhibitor complex with the lowest calculated binding energy (ΔE_{Bind}) is the closest to the native one. Thus, predicting the protein-inhibitor complex can be considered as multidimensional optimization problem. Various optimization algorithms have been proposed to solve this flexible docking problem. AutoDock3[1] (AD3) is one of the widely used docking programs of this type and utilizes a Lamarckian Genetic Algorithm (LGA) solve this optimization problem. It allows flexible docking of the ligand while treating the receptor as rigid. In AD3 the binding energy of the ligand is predicted based on Eq. 1

$$\Delta E_{Bind} = \Delta E_{vdW} + \Delta E_{elec} + \Delta E_{hbond} + \Delta E_{desolv} + \Delta E_{tors} \quad (1)$$

2 PSO@Autodock

PSO@Autodock, our fast flexible docking tool for virtual screening is based on Particle Swarm Optimization[2] (PSO) and has been implemented in AD3. It employs *var*CPSO-ls

(***v***elocity ***a***daptive and ***r***egenerative **C**onstriction **P**article **S**warm **O**ptimization with **l**ocal **s**earch) algorithm. PSO is a form of swarm intelligence and has been developed to describe the social behavior of flocking birds. If one member of the swarm (particle) sees a desirable path to go (i.e. the global minimum of ΔE_{Bind}) the rest of the particles will follow quickly even if they are on the opposite side to the particle in the multidimensional hyperspace. In *var*CPSO-ls, each particle is initialized at random position in the real valued search space. The position of the particle *i* is represented by the vector x_i = (x_{i1}, x_{i2}, ..., x_{iN}) where *N* is the dimension of the problem. The velocity of each particle is given by the vector v_i = (v_{i1}, v_{i2}, ..., v_{iN}). Furthermore, each particle has two kinds of memory that influence the movement in the search space. The cognitive memory p_i = (p_{i1}, p_{i2}, ..., p_{iN}) stores the best previous position visited by each individual particle. The social memory p_g = (p_{g1} , p_{g2}, ..., p_{gN}) contains the position of the best point in search space visited by all particles in the swarm. In each swarm move, the particle velocity is updated according to Eq. 2

$$v_i(t+1) = \chi \cdot \{v_i(t) + rand(0,1) \cdot c_1 \cdot (p_i - x_i(t)) + rand(0,1) \cdot c_2 \cdot (p_g - x_i(t))\}, \quad (2)$$

with the constriction factor $\chi = \frac{2}{2-\sqrt{\phi^2-4}}$. ϕ is computed with cognitive and social parameters c_1 and c_2, according to $\phi = c_1 + c_2$, $\phi > 4.0$.
After the velocity has been calculated, the positions of the particles are updated according to Eq. 3

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (3)$$

In *var*CPSO-ls, the constriction factor χ controls the magnitude of the particle velocity leading to a fast convergence. *var*CPSO-ls is characterized by three important features, which help to reduce the computational cost while maintaining the diversity of the swarm and preventing from premature convergence in local minima. (i) *velocity adaptive*: The velocity of the particle is updated only if the current fitness of the particle is lower than its fitness at previous iteration. (ii) *regenerative*: If a particle moves out of the search space, the particle is removed from the swarm and regenerated with a new random position. (iii) *local search*: At each swarm move, the best particle of the swarm S_{best} undergoes local search according to Solis and Wets.

3 Performance of PSO@Autodock

We accessed the performance of *var*CPSO-ls implemented in PSO@Autodock employing a suite of seven different protein-ligand complexes and compared it to the default LGA of AD3 (Table 1). The optimization process was performed with 1.5 million evaluations for LGA and 50,000 evaluations for PSO@Autodock. Fig.(1) indicates that *var*CPSO-ls of PSO@Autodock clearly outperforms the default LGA of AD3. In order to compare the performance on VS we selected the complex of HIV protease 1 with XK263 (1hvr.pdb) for screening the NCI diversity data set[3] (1990 compounds with unique scaffolds). Since PSO@Autodock uses the same objective function (Eq. 1), the correlation coefficient between the predicted ΔE_{Bind} of LGA and that of *var*CPSO-ls is 0.94.
The performance speed of PSO@Autodock is comparable to docking programs tailored for VS, like Glide (version4.0, Schrödinger Inc, San Diego, CA 92122) or GOLD (version3.0,

	LGA		*var*CPSO-ls		Top Hits in VS (*var*CPSO-ls)	
	ΔE_{Bind}	RMSD	ΔE_{Bind}	RMSD	NCI Number	ΔE_{Bind}
PDB ID	kcal/mol	Å	kcal/mol	Å		kcal/mol
3ptb	-8.15	0.32	-8.15	0.29	NCI371876	-11.34
2cpp	-7.40	1.76	-7.32	1.09	NCI101825	-10.57
2mcp	-5.45	1.09	-5.02	0.95	NCI12181	-10.99
1stp	-10.93	0.46	-10.85	0.39	NCI3354	-11.96
1hvr	-21.31	0.69	-21.34	0.29	NCI343227	-14.62
4hmg	-8.39	0.78	-8.30	0.80	NCI179187	-10.18
4dfr	-13.03	1.10	-12.64	0.96	NCI371884	-13.24

Table 1. Binding energies and rmsd values predicted by LGA and *var*CPSO-ls. Shown are also the top hit of the NCI diversity data set[3] screening employing PSO@Autodock.

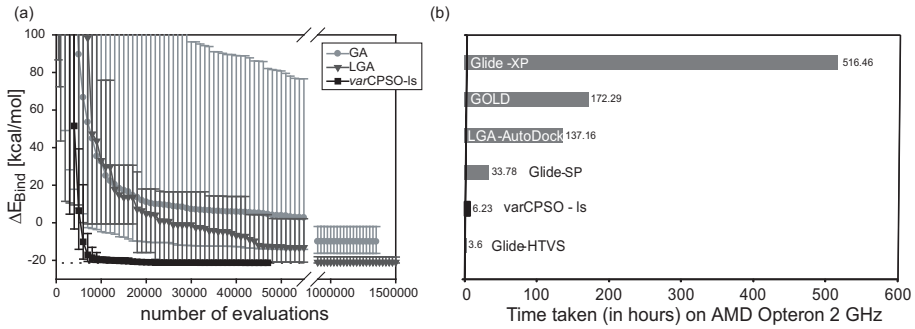

Figure 1. (a) Performance of *var*CPSO-ls and genetic algorithms(GA and LGA) on 1hvr.pdb. (b) Computing times for screening the NCI diversity data set on 1hvr.pdb.

CCDC, Cambridge UK). Thus, PSO@Autodock employing the novel *var*CPSO-ls algorithm is well suited for virtual screening of large libraries within an accurate force field approach.

Acknowledgments

We are indebted to SFB 610 for financial support.

References

1. G.M. Morris, D.S. Goodsell, R.S. Halliday, W.E. Hart, R. Belew, A.J. Olson, *Automated Docking Using a Lamarckian Genetic Algorithm and an Empirical Binding Free Energy Function*, J. Comp. Chem. **19**, 1639, 1998.
2. J. Kennedy, and R.C. Eberhardt, *Particle Swarm Optimization*, Proc. IEEE Intl. Conf. Neural Networks, Piscataway, NJ , 1942, 1995.
3. `http://dtp.nci.nih.gov/`